

I. PLAYCARD PROBLEM

A pack of cards consists of 52 cards, which have 13 different values 1,2,3,4,5,6,7,8,9,10,J,Q,K, and four different suits: Spades, Clubs, Diamonds, Hearts (Spade and Club are black, Diamonds and Hearts are red). We have a pack of full cards, which we shuffle. Pulling out successively 12 cards of the pack we create 3 rows of 4 cards each, which we place on the table.

If, in that phase, we pull out a J, Q or K we put that card at the end of the pack and continue pulling out cards until 3 rows of 4 cards each are completed. We now proceed checking if among the cards, we have put on the table, there are pairs of cards whose values give 10 as sum. If there are two cards with the value of 10 each, one of them is considered as having value 0.

On the above pairs of cards we place two cards of the pack (one on each card of the pair).

If during this phase we pull out a J, Q or K we place this on one card of the pair and this card does not take part in the procedure any more.

The procedure continues until the 12 faces of the cards are covered by J,Q,K or there are no more pairs of cards with sum 10.

Write a program which simulates the above procedure with the following demands:

1. - The number N of repetitions of the above procedure must be entered from the keyboard. When N equals 1, each of 2,3 and 4 below takes place.

2. - (a) The pack of cards must be created and, for each repetition the pack of cards must be shuffled by the program.

(b) The pack of cards must be displayed on the screen.

3. - (a) The 12 cards must be placed on the screen according to the procedure described above.

(b) The remaining cards in the pack must be displayed on the screen.

4. - (a) The cards that are covered according to the above procedure must be replaced with the cards that replace them and this must be shown on the display.

(b) After each replacement, the remaining cards of the pack must be displayed on the screen.

5. - After 5 repetitions, a histogram must be presented which will display the number of cards which remain in the pack after the end of each procedure.

Evaluation:

1 10 points

2(a)	15 points
(b)	5 points
3(a)	10 points
(b)	5 points
4(a)	10 points
(b)	5 points
5	15 points
Code	10 points
Jury	15 points

II. TREES PROBLEM

A farmer wants to preserve a rare class of ancient cypress trees. In order to do that he has taken note of the position of each tree and he has decided to surround the trees with wire drawing a polygon such that they lie entirely inside it. In order to reduce his costs, he needs to minimize the length of wire. The farmer wants to build a rectangular house, one of its sides being parallel to the X-axis, and he needs to know the relative location of the house:

- (1) The house is outside the polygon.
- (2) The house is inside the polygon.
- (3) The wire divides the house in two regions whose areas are different from zero.

Write a program capable of accomplishing the following tasks:

- (A) Finds the trees that will be the vertices of the polygon.
- (B) Calculates the length of wire that will be used.
- (C) Indicates in which of the above mentioned locations (1,2,3) the farmer's house is.

Input:

- N: The number of the trees.
- (X_i, Y_i) $1 \leq i \leq N$, $N \leq 20$, $X_i, Y_i > 0$; The coordinates of the points corresponding to each tree.
- $(a, b), (c, d)$, $a, b, c, d > 0$; The beginning and ending points of the house's main diagonal.

Output:

- A sequence of M points ($1 \leq M \leq N$) with the property that if we trace through the points in the order in which they appear, we trace the outline of the polygon.
- The length of wire that will be used.
- The indication of the position of the house in the form "1", "2", or "3".

Evaluation:

Input	10 points
A	40 points
B	10 points
C	30 points
	10 points reserved for the Jury

III. ***SQUARE PROBLEM (Selected for the 1st round)

Enumerate the position of a 5x5 matrix, in the following way: If the number i ($1 \leq i \leq 25$) has been assigned to a matrix position with coordinates (x, y) , then the number

i+1 can be assigned to the matrix position with coordinates (z,w) according to one of the following rules:

- (1) $(z,w) = (x+3,y)$
- (2) $(z,w) = (x,y+3)$
- (3) $(z,w) = (x+2,y+2)$

The problem is:

(A) Write a program that enumerates the positions of the 5x5 matrix for a given starting position with the number 1.

(B) Compute the number of all possible enumerations for every starting position at the upper right part of the matrix, including the main diagonal.

Example:

If the matrix position with coordinates (2,2) is selected as the starting position, then the next matrix position to which number 2 will be assigned, can be one of the following positions with coordinates: (2,5) or (5,2) or (4,4). These positions are marked in figure 1 by an asterisk (*).

```
      1  2  3  4  5
+---+---+---+---+---+
1 :   :   :   :   :
+---+---+---+---+---+
2 :   : 1 :   :   * :
+---+---+---+---+---+
3 :   :   :   :   :
+---+---+---+---+---+
4 :   :   :   * :   :
+---+---+---+---+---+
5 :   : * :   :   :
+---+---+---+---+---+
```

Note:

It will be appreciated if the output looks like the one in Figure 1.

Evaluation:

- (A) 50 points
- (B) 25 points
- Output 15 points
- 10 points reserved for the Jury.

IV. LANGUAGES PROBLEM

You are given an ASCII text file which contains some natural language text (eg. English, French, German, etc.) where the language used is unknown, but the characteristic of the language is the Latin style (Latin alphabet).

The problem is:

Analyze the contents of this file to determine the language used.

(A) Write a program which will read the contents of the file and count the number of characters in it. Print the total.

(B) Modify the program in (A) to also count the number of occurrences of each letter of the alphabet, converting delimiters and punctuation to the space (' ') character, and lowercase characters to uppercase ones, i.e. the only

characters considered in the count will be elements of the set [' ','A'..'Z'].

Sort the characters in this set in order of frequency, i.e. the most frequently occurring characters appearing at the start of the list. Print the sorted list out.

(C) Modify the program written in (B) to count the frequency of characters. Normalise the counts, i.e. divide the frequency of each character by the total number of characters read; this will give you a relative frequency which is independent of the number of the characters in the text. Write the relative frequency counts to a data file.

(D) Extend the program written in (C) so that it may accept to read a text data file and compare it with given text files of known languages. The comparison method will be of your own invention. The result should be a report of the language which the program thinks the original text is written in.

Note:

The given text files of the known languages are under the names ITA, FRA, etc. for Italian, French, etc. respectively. The text data file has the name TEXT.

Evaluation:

- (A) 10 points
 - (B) 20 points
 - (C) 20 points
 - (D) 40 points
- 10 points reserved for the Jury.

Sample text:

Morire in conseguenza di anabolizzanti intrapresa per migliorare le proprie prestazioni atletiche 'e poi molto diverso dal morire al volante di un'auto di formula uno mentre si tenta di battere un record affrontando i rischi legati a quelle prove, o dal perire vittime di una scarica di sassi mentre si tenta una prima salita su una parete Nord di qualche colosso alpino? Che ci sia o no di mezzo la farmacologia, l'artificiale, il chimico, conta relativamente poco, sul piano essenziale, anche se sembra molto rilevante dal punto di vista dell'immaginario cillettivo; il quale ancora considera spesso un eroe il pilota di auto da corsa o il grande alpinista, e ha invece un atteggiamento molto diverso nel caso di chi perisca vittima di un doping incauto. Naturalmente , qui entrano in gioco altri elementi, solo indirettamente morali. Fare uso di stimolanti chimici 'e in genere vietato nello sport, chi viola questa norma commette una grave slealtà: ma qui l' esecrazione morale non 'e motivata anzitutto dal fatto che viene messa a repentaglio l'integrità fisica dell'atleta, bensì dal mancato rispetto per regole del gioco. Le quali, per altro, vietano il doping anche e soprattutto per il danno fisico che esso alla lunga provoca; ma in base allo stesso principio, forse, dovrebbero vietare anche pratiche sportive "leali" e tuttavia non meno pericolose per chi le pratica..

An s-term is a sequence of S's and parentheses defined recursively as follows:

S is an s-term, and if M and N are s-terms, (MN) is also an s-term.

Example of an s-term:

((((SS)(SS))S)(SS))

The right parentheses provide no new information, so they can be omitted, i.e. (MN instead of (MN)), so that the previous s-term becomes:

((((SS(SSS(SS

1. Write a procedure 'gensterm' to generate s-terms: your procedure should generate n (n=length=number of S's) textfiles containing all s-terms of length 1,...,n respectively. S-terms should be separated by ";". The end of the last s-term in each file should be marked with ".".

Write a program that accepts an integer n(<=10), uses the above procedure and displays all generated s-terms on the screen.

Consider a calculus with s-terms. The only algebraic rule (s-rule) that can be used is the following: any subterm of an s-term that has the form ((SA)B)C (where A,B and C are s-terms) can be rewritten as: ((AC)(BC)) i.e.

Context1(((SA)B)C)Context2->Context1((AC)(BC))Context2

The application of this rule on an s-term is called 'reduction' of the s-term.

There are different ways (strategies) of choosing a subterm to apply the s-rule. The successive application of the s-rule on an s-term until no more applications of the s-rule are possible is called 'normalization' of the s-term.

Example of a reduction chain:

((((SS)(SS))S)(SS)) -> (((SS)((SS)S))(SS)) ->

((S(SS))(((SS)S)(SS))) -> ((S(SS))((S(SS))(S(SS))))

2. Choose an appropriate data structure for representing s-terms which can facilitate the application of the s-rule. Write two procedures 'readterm' and 'printterm' that transform s-terms to (and from) your representation from (and to) the form generated by 'gensterm'. Your program should be able to demonstrate these transformations.

3. Write a procedure 'reduce' to perform one reduction step according to the s-rule on a specified subterm of an s-term given in your representation. Your program should be able to demonstrate this.

4. Write a procedure 'normalize': given an s-term, it should repeatedly choose a subterm to apply the s-rule, until no further reductions are possible or until the number of reduction steps exceeds some maximum, e.g. 30. Your program should be able to demonstrate this.

5. Finally, incorporate all of the above in a program that:

- a) requests a length n from the user,
- b) uses s-terms of length n, generated by 'gensterm',
- c) transforms s-terms into your representation,
- d) normalizes (if possible),
- e) outputs the resulting (normalized) s-terms,
- f) outputs the number of reduction steps performed for each s-term, or a 'not normalized' message in case of unsuccessful normalization in 30 steps, and
- g) outputs the number of not normalized vs. the total number of s-terms for the given length n.

EVALUATION:

- (1) 20 points
- (2) 25 points
- (3) 15 points
- (4) 20 points
- (5) 10 points
- Jury 10 points (clarity, elegance, style)

 VI. MAXIMUM GANG

A Police captain knows well all the outlaw persons of his city, as well as, every possible collaboration among them. He would like to find the maximum gang of the city.

In this case, a gang is a subset of outlaw persons so that any person in it collaborates with all others in the subset. Maximum gang means that there does not exist another gang with greater cardinality.

Create a program capable of carrying out the following tasks:

- (A) Accept the police captain's data with a total number of outlaw persons less than 41. Use as input data file an ASCII one with the following structure:

```

a(1,1)
a(2,1)a(2,2)
a(3,1)a(3,2)a(3,3)
.....
a(n,1)a(n,2)a(n,3).....a(n,n)

```

where $a(i,j)=1$, if person i is collaborating with person j or $i=j$, and $a(i,j)=0$, otherwise. For instance (in the case of 6 persons):

```

1
01
101
1011
01101
101111

```

In this example an output can be the following one:

```

A Maximum Gang is :
1 3 4 6
cardinality = 4

```

- (B) Extend the input part of the program, so that to generate data in a random way under a given probability $0 < d < 1$ of collaboration of outlaw persons.
- (C) Using random data or input file, find the maximum gang of the city. Use the same output format as in the example above (see task A).

EVALUATION

- (A) 5 points
 - (B) 15 points
 - (C) 50 points
 - Total Machine Time 20 points
 - Jury 10 points
-

VII. MEDICAL VISITOR'S REGISTER

A medical visitor would like to have a program which would schedule all his medical appointments and help him meet as many doctors as possible in one day in order to advertise the medical products of the company he represents.

You are asked to write the program which would do this task for him. The input file consists of lines, each of which contains the name of a doctor, the beginning and the end of the interval when the doctor is ready to meet the medical visitor, as well as the name of his medical institute.

The rules for scheduling the appointments are as follows:

1. An appointment has at least 70 minutes duration; furthermore the medical visitor needs at least 30 minutes between any two appointments to travel to the next medical institute.

2. All the times in the input file belong to the same day, and the medical visitor does not want to meet any doctor twice on the same day.

3. Also, two consecutive appointments cannot be held at the same institute.

4. Among several doctors the medical visitor always prefers the one whom he can meet earlier.

5. If there is more than one doctor whom the medical visitor would be able to meet at the same time (either because by the time he finishes from the previous appointment there are doctors already waiting or because their proposed appointment would begin later but at the same time during the day) he prefers the one who has less remaining time for the given day (but of course this time at the beginning of the appointment must be enough for the required 70 minutes).

6. To visit as many doctors as possible, the medical visitor always (even during an appointment) keeps in mind the next appointment's starting time, therefore if it is necessary he can terminate the on-going appointment after the minimal 70 minutes expire, take his 30 minutes for traveling and begin the next appointment according to rule 5.

7. If he is not in a hurry for a new appointment (according to rule 6), the medical visitor stays with a doctor as long as he can.

Input file: consists of lines each of which contains a name (with the English alphabet), a blank space , a time (in hh.mm form) indicating the possible beginning of an appointment, a dash ('-'), a time again indicating the last possible moment of the appointment, a blank space, and a name (with the English alphabet) of the medical institute.

The input file does not contain empty lines, nor is the end of it marked with special characters.

A possible input file is as follows:

Bob 16.00-17.25 Cross
John 09.30-11.50 Health
Charles 11.00-20.00 Chest
Don 08.00-13.20 Cross
Norman 22.00-23.05 Brain
Jerry 10.00-17.00 Health
Charles 09.20-10.40 Orthopedic
Evelyn 19.15-20.40 Orthopedic
Peter 09.35-11.55 Brain
Don 18.00-20.00 Eye

Output file: should contain a table in which all the possible appointments appear in their proposed chronological order. The appointments which are ruled out by any of the rules above should not appear in the table. The table should consist of lines containing the realizable beginning and end time of the possible appointments, their place, and the proposing doctor's name. The exact spacing of the output table is not important, but should look similar to the one below. The output for the previous input file is as follows:

08.00-09.10	Cross	Don
09.40-10.50	Health	John
11.20-12.30	Chest	Charles
13.00-15.30	Health	Jerry
16.00-17.25	Cross	Bob
19.15-20.40	Orthopedic	Evelyn

8. Solve the above problem for two medical visitors A and B preserving the same scheduling rules and the additional restriction that they are not allowed to visit the same doctor. Each time an appointment is available, it is taken by that medical visitor who has been free for the longest period. If both have been free for the same time period, Mr. A takes the appointment. The output consists of two appointment lists.

Evaluation:

One medical visitor	45 points
Two medical visitor	45 points
Jury	10 points (elegance, clarity,
progr. style)	